

在单 202012707

一、

1. 将程序中的逻辑地址转换为内存中的物理地址的过程
2. 在指令运行时, 实际地址转换 (相对地址转换为绝对地址)
3. 具有请求调入功能和置换功能, 能从逻辑上对内存容量加以扩充的一种存储管理系统
4. 在程序运行前, 将各目标模块及它们所需的库函数链接成一个完整的装配模块, 以后不再拆开, 对相对地址进行修改, 要替换外部调用符号
5. 把内存中暂时不能运行的进程或暂时不用的进程和数据, 调出到外存上, 以便腾出足够的内存空间, 再把已具备运行条件的进程或进程所需的程序或数据调入内存
6. 将以上层软件发来的抽象 I/O 请求, 再把它转换为具体要求后, 发送给设备控制器启动设备去执行
7. 外围操作与 CPU 对数据的处理同时进行, 在联机情况下实现的同步操作
8. 一种特殊的处理机, 具有执行 I/O 指令的能力, 并能通过执行通道程序控制 I/O 操作
9. 是 OS 的一部分, 它提供某种管理机制, 以便 OS 对系统及所有用户的数据与程序进行存取保护和访问
10. 把源程序经过编译程序编译后, 但尚未经过链接程序链接的目标代码"所构成的文件"
11. 从用户观点所观察的文件组织形式
12. 在记录式文件中, 每个记录都用于描述实体集中的一个实体, 各记录有着相同或不同数目的数据项
13. 利用二进制的位来表示磁盘中一个盘块的使用情况
14. 操作系统提供给应用程序或其他系统软件的一组函数、宏、数据符号等的集合, 用来描述应用程序与操作系统之间的交互通信
15. 应用程序通过系统调用间接调用 OS 中的 I/O 过程, 对 I/O 设备进行操作
16. CPU 对 I/O 设备发来的中断信号的一种响应
17. 是 OS 的一部分, 它提供某种管理机制, 以便 OS 对系统及所有用户的数据与程序进行存取保护和访问
18. 具有文件名的一组相关关系的集合
19. 从用户观点所观察的文件组织形式
20. 文件的存储结构, 指系统将文件存储在外存上所形成的一种存储组织形式

二、

1. ~~地址映射~~ 动态重定位
2. 绝对装入方式 可重定位装入方式 动态运行时装入方式
3. 静态链接 装入时动态链接 运行时动态链接
4. 空闲分区表 空闲分区链

5. 紧凑

6. 固定系统 不固定 程序

7. 对换性 虚拟性

8. 文件管理接口 文件管理软件 文件存储设备

9. 分区分配 分区回收 碎片整理

10. 外部碎片

11. 分页 分段 段页式

12. 快表

13. 逻辑地址 物理地址

14. 便于共享 便于动态链接 便于保护信息保护 便于信息共享 便于动态增长
方便编程

15. 地址变换机构

16. 局部性 时间局部性 空间局部性

17. 处理机 指令类型单一 与CPU并行工作

18. 顺序文件 索引文件 索引顺序文件

19. 单级索引 多级索引 混合索引

20. 动态重定位

21. 用户级I/O软件 设备独立性软件 设备驱动程序

22. 顺序结构 链接结构 索引结构 索引结构

23. 高*速缓存 缓存 外存

24. 连续分配 离散分配

25. 存储介质特性 存取方式

26. 顺序文件 索引文件 哈希文件

27. 连续分配 链接分配 索引分配

28. 绝对路径 相对路径

29. 空闲盘块链 空闲盘区链

30. 基于索引节点的共享 基于有连接链的共享

31. 块设备 字符设备 独占设备 共享设备 虚拟设备

32. DMA方式 通道控制方式

33. 单缓冲 循环缓冲 缓冲池

34. 寻道时间 传输时间

三、

1. 高速缓存: SRAM, 速度快, 容量小, 存放CPU近期访问的数据

内存: DRAM, 速度较快, 容量较大, 存放运行中的程序和数据

外存: 磁盘, SSD等, 速度慢, 容量大, 长期存储数据

速度: $Cache > 内存 > 外存$ 容量: $外存 > 内存 > Cache$ 成本: $Cache > 内存 > 外存$

2. 编译与源代码: 程序员编写高级语言代码

编译: 编译器将源代码转换为目标代码

链接: 链接器将目标代码与库函数结合, 生成可执行文件

装入: 操作系统将可执行文件加载到内存

运行: CPU执行程序指令, 处理数据

终止: 程序执行完毕, 释放内存资源

3. 静态链接, 装入时动态链接, 运行时动态链接

① 运行速度快, 独立性高, 占用空间大, 更新困难

② 节省空间, 便于共享库更新, 启动时间较长

③ 灵活性高, 节省内存, 管理复杂, 可能影响性能

4. 首次适应: 从内存的低地址开始查找第一个足够大的空闲分区

最佳适应: 选择最小的能满足需求的分区, 以减少碎片

最坏适应: 选择最大的空闲分区, 以减少碎片

邻近适应: 从上次分配的位置继续查找, 提高搜索效率

5. ① 首次适应, 最佳适应, 最坏适应 ② FCFS, SJF, RR

共同: 均涉及资源分配策略, 优化系统效率

异: 内存管理关注空间管理, CPU调度关注时间分配

内存管理解决碎片问题, CPU管理解决响应时间和吞吐量问题

6. 页面: FIFO, LRU, OPT, Clock

CPU: FCFS, SJF, RR, 优先级调度

同: 均基于某种策略选择最优目标, 以提高系统效率

异: 页面: 关注内存利用率, 减少缺页率

CPU: 关注公平性, 响应时间, 吞吐量

7. 页面: FIFO, LRU, OPT, Clock

磁盘: FCFS, SSTF, SCAN, CSCAN

同: 均通过优化访问顺序减少延迟

异: 页面: 基于时间局部性, 目标是减少缺页中断

磁盘: 基于空间局部性, 目标是减少磁头移动时间

8. 分段: 通过页表查询页号对应的物理块号, 与偏移量组合成物理地址

分段: 通过段表查询段基址和段长度, 检查偏移合法性, 基址+偏移=物理地址

分段: 先通过段表找到页表基址, 再通过页表找到物理块号, 最后与偏移组合

9. 中断请求: 设备或异常触发中断信号

中断响应: CPU保存当前现场, 关闭中断

中断判优: 识别中断源并确定优先级

中断处理: 执行对应的中断服务程序

中断返回: 恢复现场, 重新开放中断, 继续原程序

10. 用途: 系统文件, 用户文件, 库文件

数据类型: 文本文件, 二进制文件

组织方式: 顺序文件, 索引文件, 哈希文件

管理方式: 普通文件, 目录文件, 设备文件.

11. 单级目录: 所有文件在同一目录下, 查询简单但易重名.

二级目录: 分主目录和用户目录, 解决重名问题

树形目录: ~~多级目录~~ 多级层次结构, 支持绝对路径和相对路径查询

图形目录: 通过硬链接或符号链接实现文件共享, 查询需防循环.

12. 顺序结构: 优点: 连续存储, 访问速度快.

缺点: 增删困难, 易产生碎片

链接结构: 优点: 动态增长, 无外部碎片.

缺点: 随机访问效率低

索引结构: 优点: 支持随机访问和动态增长.

缺点: 索引表占用额外空间

13. 内存: 连续分配: 固定分区, 动态分区

离散分配: 分页, 分段, 段页式.

外存: 连续分配: 文件占用连续磁盘块

链接分配: 通过指针链接非连续块

索引分配: 通过索引表记录块.

统一: 均需管理空闲空间, 解决碎片问题.

异: 内存分配更注重速度, 外存分配更注重容量和持久性.

14. 磁盘调度算法: 如 SSTF, SCAN 等减少寻道时间.

缓存技术: 使用磁盘缓存暂存频繁访问的数据.

RAID技术: 通过并行读写提高吞吐量.

提前读: 预取相邻数据块.

延迟写: 合并多次写操作, 减少 I/O 次数.